# Controlling Parasitics in Linear Optical Processors

I. Zand, B. Abasahl, U. Khan, and W. Bogaerts

[1] Ghent University - IMEC, Photonics Research Group, Department of Information Technology.
Technologiepark-Zwijnaarde 15, 9052 Gent, Belgium
[2] Center of Nano- and Biophotonics (NB Photonics). Ghent University, Belgium

*Parasitics of programmable waveguide meshes can induce chaotic subcircuits of randomly coupled ring resonators. We investigate limitations and possible strategies to improve the response of the programmed circuit. Findings show that setting unused couplers to cross-state considerably reduces the impact of the parasitics errors.*

## Introduction

Programmable *photonic integrated circuits* (PIC) have gained a lot of interest in the last few years [1-6]. Compared to photonic circuits designed for a specific function, programmable PICs can be configured for a variety of functions. At their core we find a mesh of waveguides and tunable 2×2 couplers, where the connectivity can be defined in both amplitude and phase. In such a mesh, the couplers can be configured to perform routing functions, or implement interferometers and resonators. There are feed-forward meshes, connecting a set of inputs to a set of outputs [1, 2], or meshes with loops, which can connect all waveguide ports to each other [3-6]. Both types perform a linear transformation on optical signals, and are called linear optical processors. Essential to these meshes are electrically controlled 2×2 couplers and phase shifters. Their state defines the circuit, so the control scheme should be as accurate as possible. In reality, this happens in software through an electrical driver circuit. Effects such as nonlinear response curves, electrical and thermal crosstalk, or discretization in a digital-to-analog conversion can introduce imperfections in this actuation. In this paper, we studied the effect of small stochastic imperfections in the phase and coupling control in a 7-cell hexagonal linear processor.

## Methodology

To study the effect of stochastic imperfection in a linear programmable photonic mesh, we used the Monte-Carlo technique with frequency-domain circuit simulations. We extended the photonic design framework IPKISS by Luceda Photonics with automatic circuit generation for hexagonal mesh circuits that can be simulated with the Caphe circuit simulator. All parameters from the couplers and phase shifters can be individually addressed, which allows us to ’reprogram’ the circuit, but also introduce imperfections. We used a 7-cell hexagonal mesh as proposed in [5], and programmed connection routes of different lengths. We then applied normally distributed imperfections in the coupling $\kappa$ and phase shifts $\phi$ with different standard deviations $\sigma_\kappa$ and $\sigma_\phi$ , and ran Monte-Carlo cycles with 100 samples, comparing the results with the ideally programmed circuit. Our step-by-step simulation process has been illustrated in the Fig. 1. In our illustrations, we used error bars to show intensity spread of the transmission spectra for the configured paths. For this, the [*min, mean, max*] points of the error bars are obtained by

- $\min([\min(T(\lambda))_5, \ldots, \min(T(\lambda))_{95}])$
- $\mathrm{mean}([\mathrm{mean}(T(\lambda))_5, \ldots, \mathrm{mean}(T(\lambda))_{95}])$
- $\max([\max(T(\lambda))_5, \ldots, \max(T(\lambda))_{95}])$

The subscript 5 and 95 indicate that we only considered the 5-95th percentile of the samples, discarding the most extreme values.

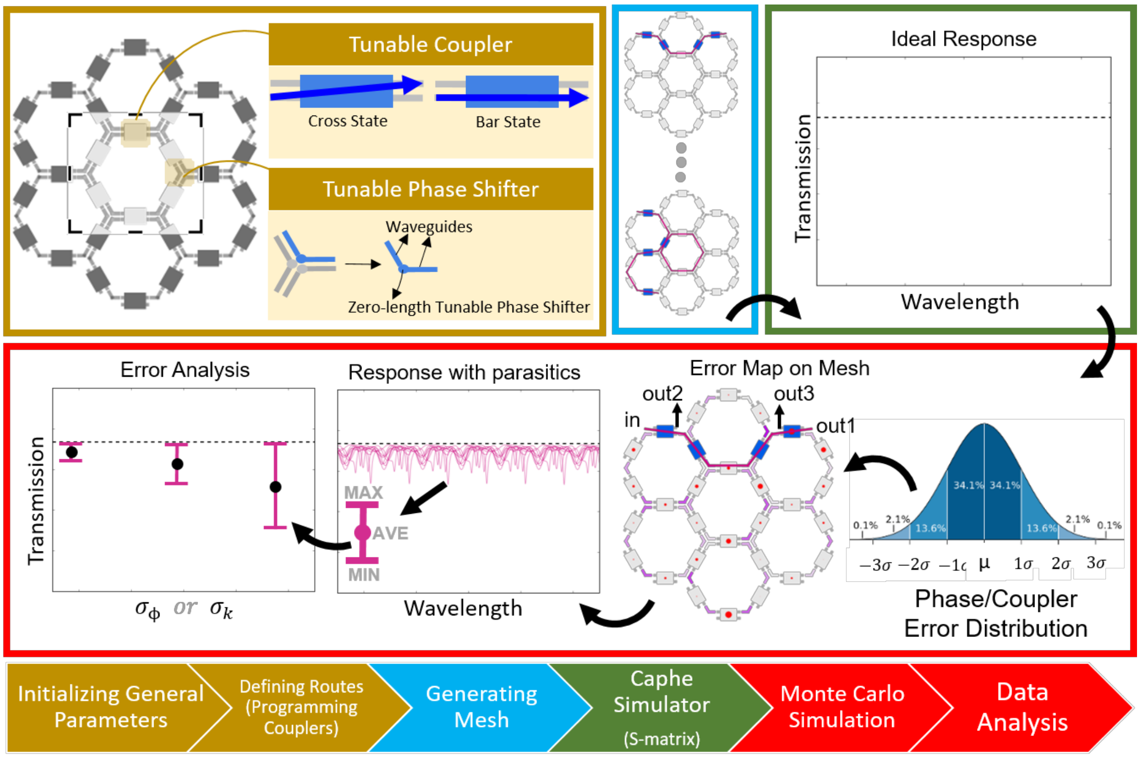

Figure 2. Simulation process of 7-cell hexagonal linear processor to study programmed meshes with parasitic components.

## Results

When routing paths through a mesh, the tunable couplers are either programmed by default in *bar* or *cross* state, so we refer to these default states as *bar-state bias* (**BS-bias**) and *cross-state bias* (**CS-bias**), respectively. Paths with length of $N$ segments also have been referred to as $P_N$. Fig. 2 shows an example of a path with the length of 13 segments ($P_{13}$) routed through a 7-cell mesh for two cases of BS-bias and CS-bias. When the mesh is ideally programmed, we expect that only the length of the path will contribute losses (black dashed lines). But as the graph in Fig. 2 shows, when we introduce random variations to the couplings $\kappa$ with a $\sigma_\kappa = 1\%$ and to the phase shifts $\phi$ with a $\sigma_\phi = 17^o$, significant ripples appear on the desired output spectrum (*out1*: red lines), and light coming out of many unwanted ports (*out2*: green lines and *out3*: blue lines). As seen, unwanted coupling will introduce additional losses, because light is tapped out of the main path, but it also introduces parasitic interference paths, and even ring resonators (periodic ripples in the red curves in Fig. 2(a)).

Comparing Figs. 2(a) and (b) shows that programming the mesh in CS-bias will considerably improve the circuit performance. It significantly reduces the fluctuations in the desired output (*out1*) and also in undesired outputs (*out2*). Moreover, it compensates light accumulation in the mesh by breaking undesired rings; comparing transmission in *out3* (blue lines) for both cases shows this. As seen, CS-bias reduces the output in *out3* with ~20dB, which indicates most of the light is propagating counter clockwise in the cell where the desired output is defined.

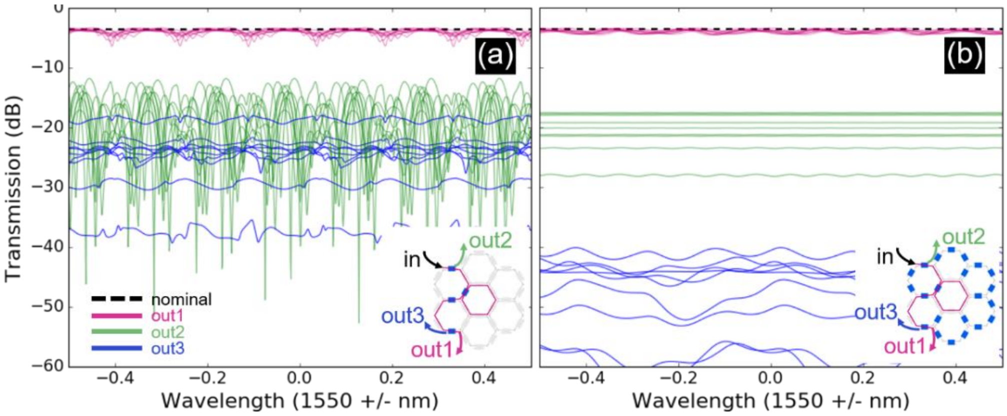

Figure 2. Transmission spectra for a 13-segment path ($P_{13}$) through a 7-cell mesh. Random phase errors with a $\sigma_\phi$ = 17° and random coupling with $\sigma_\kappa$ = 1.0% are applied for two differently programmed meshes: a) Bar-State bias (BS-bias) and b) Cross-State bias (CS-bias). Blue and gray couplers are in cross and bar states. Black dashed line corresponds to the ideally programmed meshes. Also, red lines belong to the *out1* (desired output), and blue and green lines belong to the *out2* and *out3* (unwanted outputs).

As part of a systematic study, several paths ($P_5$, $P_6$, $P_{13}$, $P_{23}$) with both simple and complex routings are configured, and the results are compared in the Fig. 3. For this mean, random phase variation of 17° and random coupling variations of 0.05%, 0.4%, and 1.0% are applied for the Monte-Carlo simulations with 100 samples. To illustrate and compare the intensity spread in the transmission spectra of BS-bias and CS-bias, red and green bars are used in each subplots, respectively. The response of ideally programmed meshes are also shown by the black dashed lines.

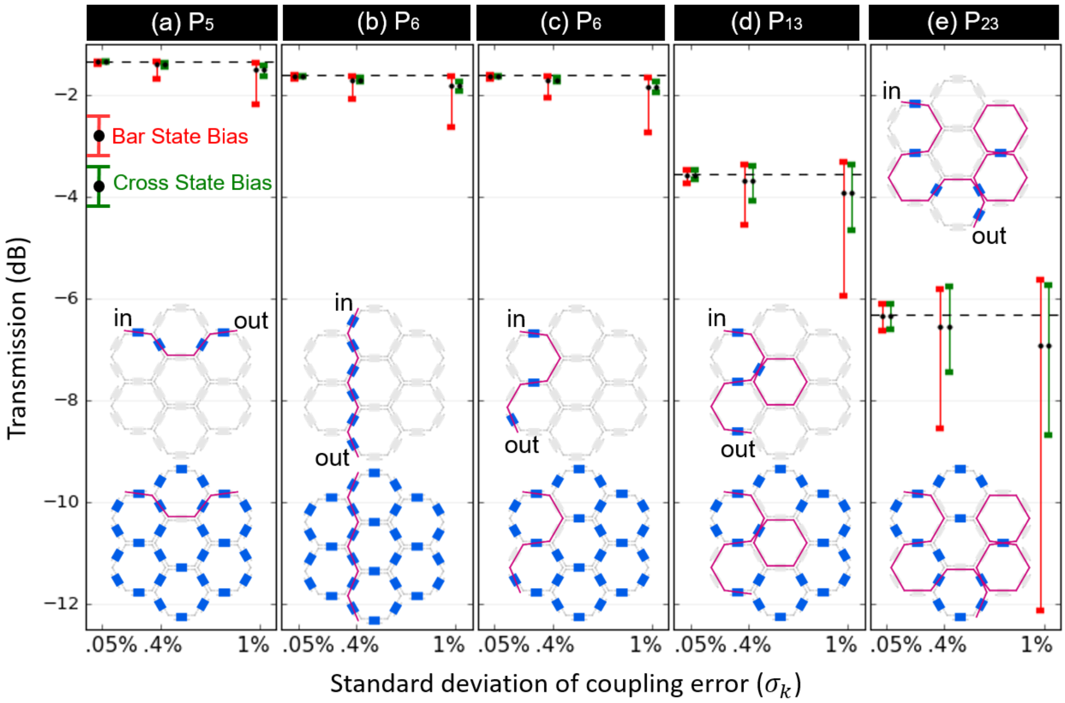

Figure 3. Spread of the transmission for different $\sigma_\kappa$ (0.05%, 0.4%, 1.0%): a) $P_{15}$ b) $P_6$ with simple routing c) $P_6$ with curved routing d) $P_{13}$ c) $P_{23}$. Intensity spread in the transmission spectra of BS-bias (Bar-State bias) and CS-bias (Cross-State bias) meshes are illustrated by the red and green bars, respectively. Also, both types of the programmed meshes are shown in the insets.

As seen in the Fig. 3, increasing length of the paths introduces more losses and makes them more vulnerable to large coupling variations. For example, $P_5$ has nominal transmission of -1.33 dB, and its average transmission drops from -1.34 dB to -1.5 dB for coupling variation of 0.05% to 1.0% (Fig. 3(a)); on the other side, $P_{23}$ has nominal transmission of -6.3 dB, and its average transmission drops from -6.33 dB to -6.9 dB (Fig. 3(e)).

Another observation is that parasitics, in complex routes ($P_{13}$, $P_{23}$), may cause higher transmissions in the desired output compared to the nominal situation (Figs. 3(d, e)); this can be explained by the fact that the transmission variations can introduce both 'loss' and 'gain' to the main path: because the main path winds through the mesh, the parasitic coupling introduces shortcuts that lower the losses when they interfere constructively. For instance, although $P_{13}$ (Fig. 3(d)) has been configured only by adding a loop to the $P_6$ (Fig. 3(c)), it has larger intensity spread in its transmission spectra. Also, in the Fig. 3(d), we see that the intensity spread in the transmission spectra can grow as large as 7dB with coupling errors of only 1.0% throughout the mesh.

Comparing green and red bars in Fig. 3, suggests that programming the mesh in CS-bias can reduce intensity spread of the spectra about 50%-80%; this reduction depends on the complexity of the paths. It is also seen that complexity of the short paths without loops ($P_6$ in Fig. 3(b, c)) does not affect their performance.

## Conclusion

We show that even small imperfections in programmable circuits can quickly deteriorate the performance of programmable circuits. Small errors (order of 1%) in coupling and phase, introduced by fabrication, driving schemes or crosstalk, can induce unacceptable transmission fluctuations. This imposes requirements on the perfection of the components and the driving schemes, the circuit topologies but also the programming strategies to implement functionality. We also showed that by proper programming of the unused couplers, the intensity spread on the transmission spectra can be reduced by 50%-80%.

## Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780283 (MORPHIC) and from the European Research Council through grant agreement No 725555 (PhotonicSWARM).

## References

[1] D. A. B. Miller "Self-configuring universal linear optical component," Photon. Res. 1(1), 1-15 (2013).

[2] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley "Optimal design for universal multiport interferometers," Optica. 3(12), 1460-1465 (2016).

[3] L. Zhuang, C. G. H. Roeloffzen, M. Hoekman, K.-J. Boller, and A. J. Lowery, "Programmable photonic signal processor chip for radiofrequency applications," Optica 2(10), 854–859 (2015).

[4] J. Capmany, I. Gasulla, and D. Perez, "The programmable processor," Nat. Photonics 10(1), 6–8 (2016).

[5] D. Pérez, I. Gasulla, L. Crudgington, D. J. Thomson, A. Z. Khokhar, K. Li, W. Cao, G. Z. Mashanovich, and J. Capmany, "Multipurpose silicon photonics signal processor core," Nature Comm. 8(636), 1–9 (2017).

[6] D. P´erez, I. Gasulla, and J. Capmany, "Programmable multifunctional integrated nanophotonics - Review Article," De Gruyter - Nanophotonics 2018 7, 1351–1371 (2018).